Basic version of WineQualityMLP (Saved):

No normalization
Layers: [11, 5, 1]
Without dropout
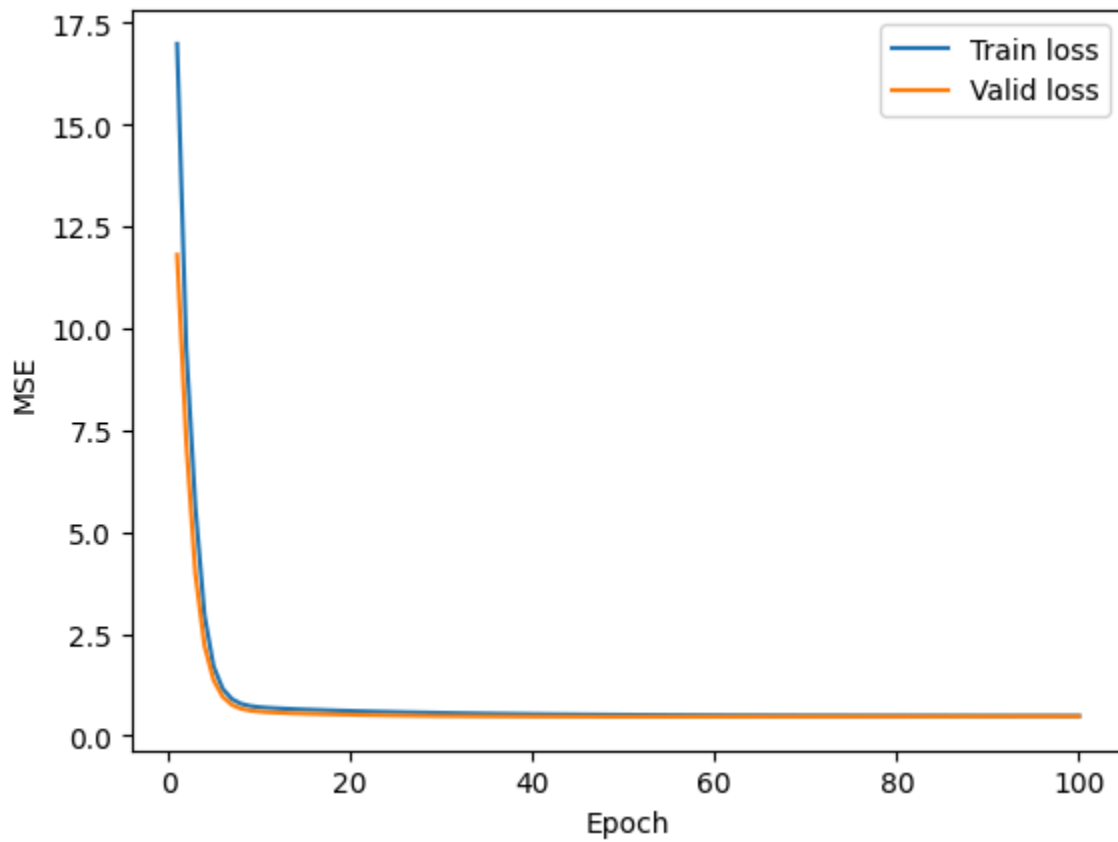**Activation function: reLu**
Loss function: MSE
Number of epochs:100
Batch size: 8
Learning rate: 0.0001

Loss curve:

The MSE of test set of the basic version of MLP: 0.5035768151283264

Tuning the model:

Hyperparameters can be tuned:

1. Normalization
    a. Basic version: no normalization
    b. Try normalize x (Standard Scaler works the best)
    c. Try normalize both x values and y values
    d. Thy batch norm
2. Shuffle data when load
    a. Basic version: no shuffle
    b. Try with shuffle
1. Number of hidden layers and neurons per hidden layer
    a. Basic version has [5]
    b. Try [5, 5]
    c. Try [10]
    d. Try [10, 2]

2. Activation functions of each layer:
    a. Basic version has relu
    b. Try Sigmoid
    c. Try Tanh
3. Number of epochs
    a. Basic version has 100 epochs
    b. Try early stopping
4. Learning rate:
    a. 0.0001
    b. 0.001
    c. 0.01
5. Dropout rate:
    a. 0
    b. 0.1
    c. 0.5
6.

Tune V1.1(Saved):

Normalization on x: <mark>StandardScaler()</mark>
Layers: [11, 5, 1]
Without dropout
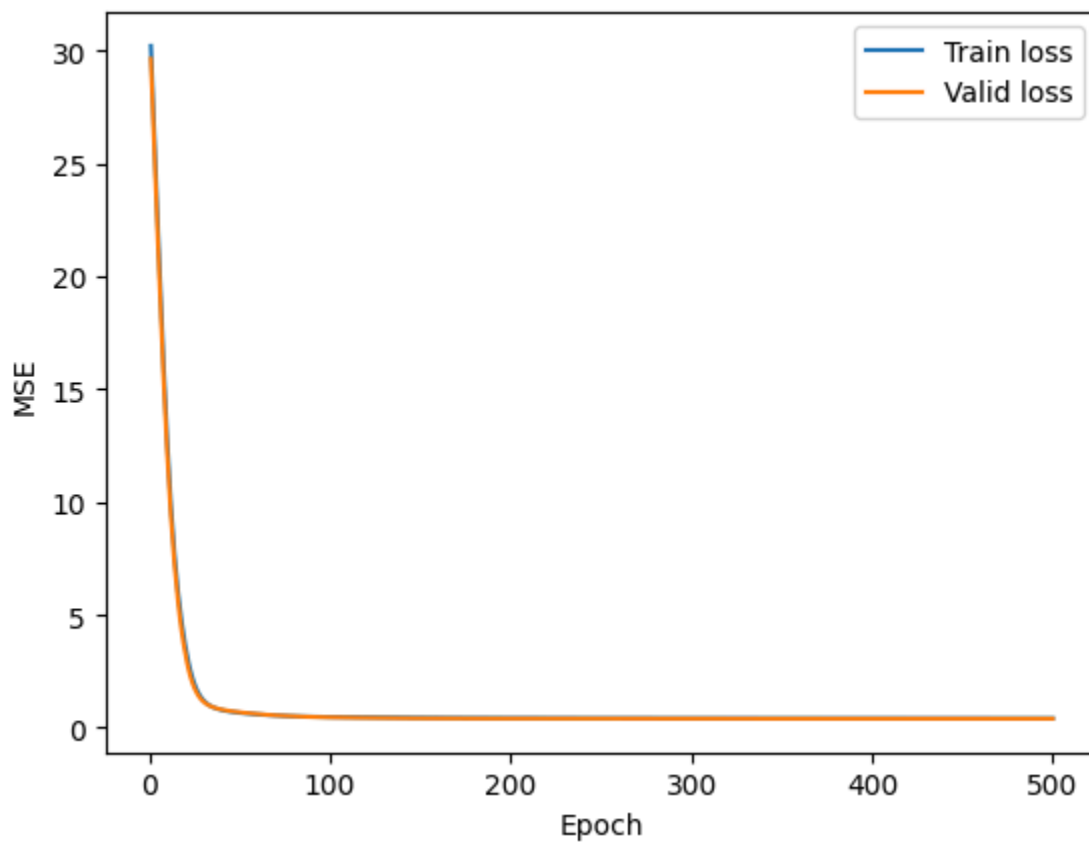Activation function: reLu
Loss function: MSE
Number of epochs: 500
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss curve:

The MSE of test set: 0.42102354764938354

Tune V1.2:

Normalization on x: MinMaxScaler()
Layers: [11, 5, 1]
Without dropout
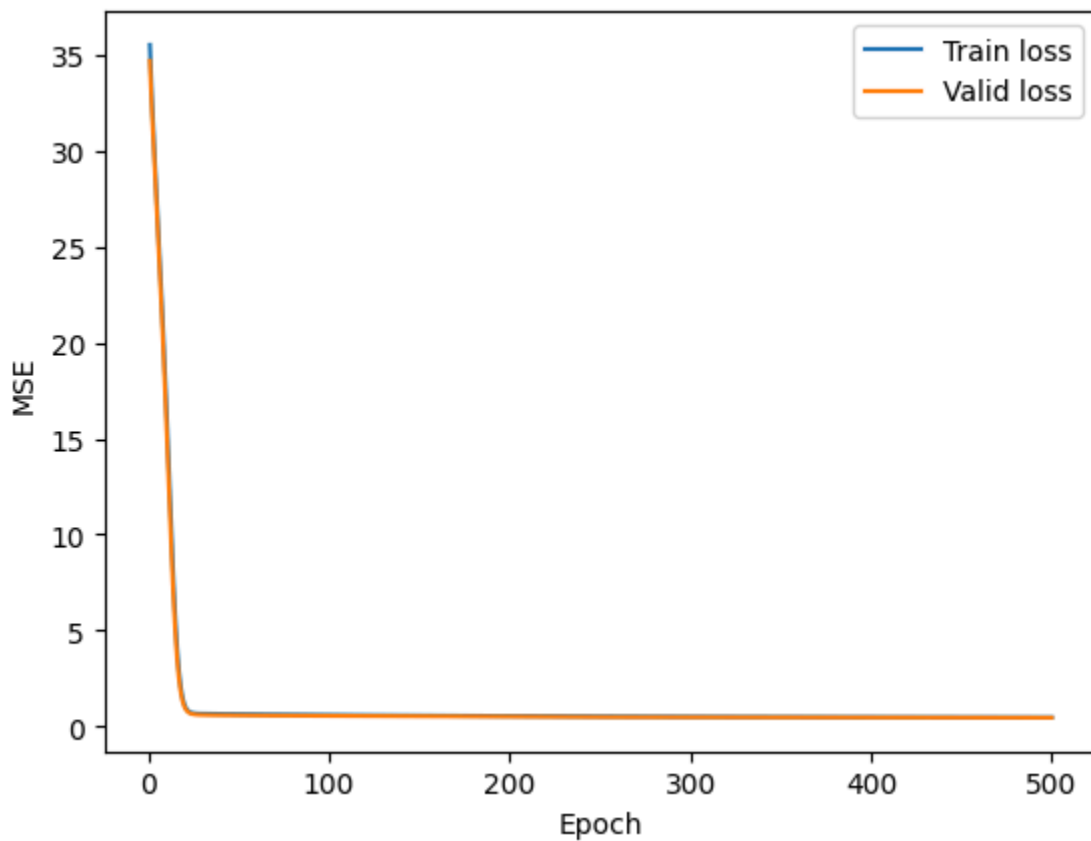Activation function: reLu
Loss function: MSE
Number of epochs: 500
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss curve:

The MSE of test set: 0.4620991349220276

MinMaxScaler can slightly improve the performance, but not as good as the Standard Scaler.

Tune V1.3(Saved):

Normalization on x: StandardScaler()
Layers: [11, 5, 5, 1]
Without dropout
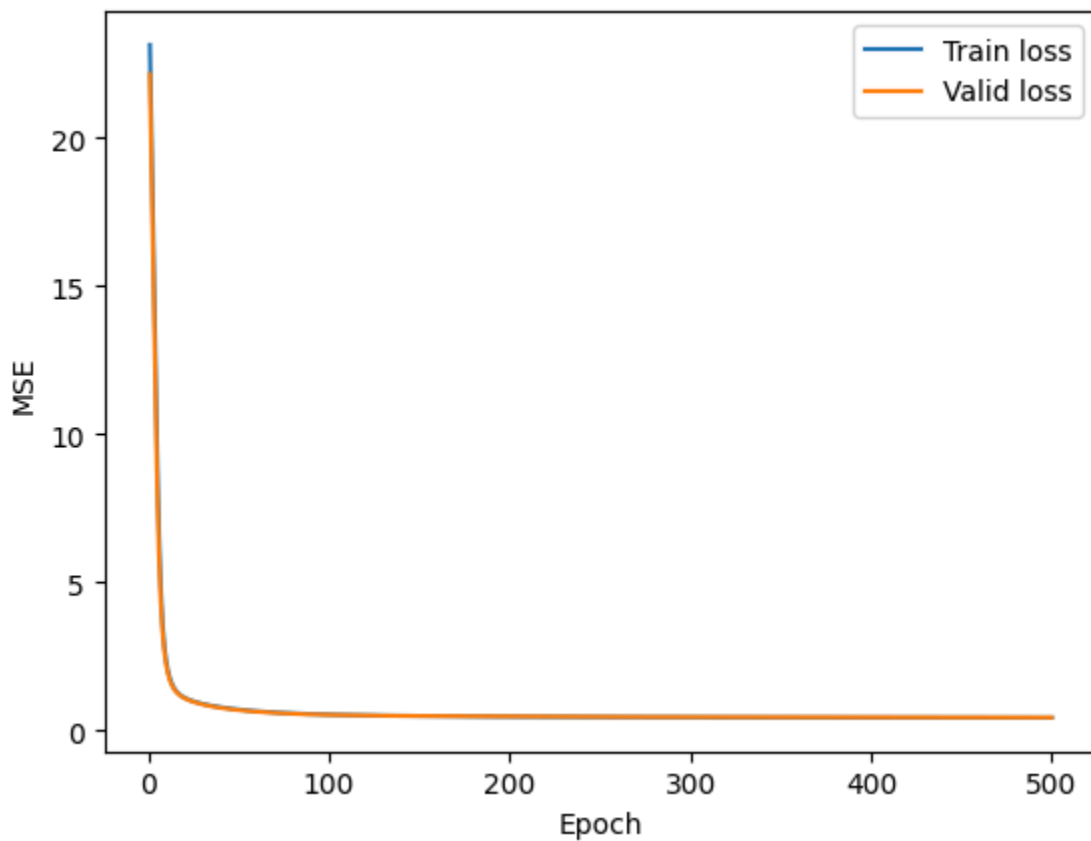Activation function: reLu
Loss function: MSE
Number of epochs: 500
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader
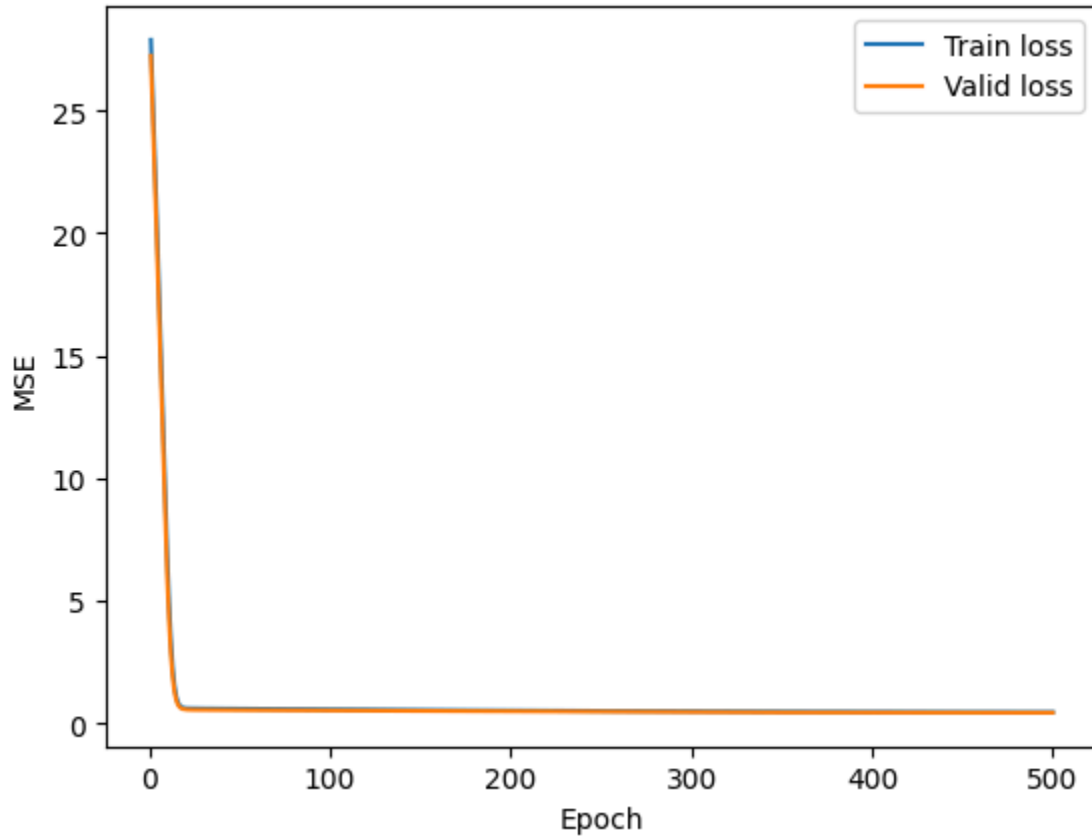

Loss curve:

The MSE of test set: 0.41365745663642883

TuneV1.4

Normalization on x: MinMaxScaler()
Layers: [11, 5, 5, 1]
Without dropout

Activation function: reLu
Loss function: MSE
Number of epochs: 500
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss curve:

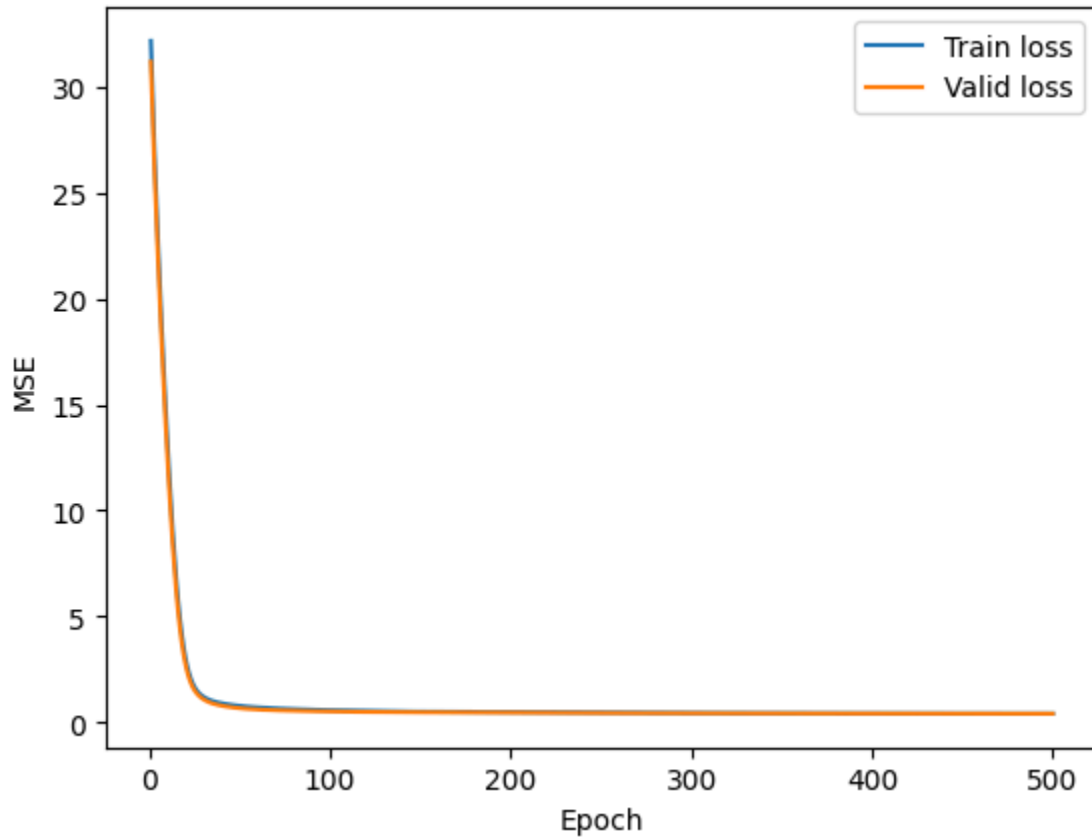**The MSE of test set: 0.4536188840866089**

This is similar to V1.2

TuneV1.5(Saved)

Normalization: SS
Layers: [11, 10, 1]
Without dropout
Activation function: reLu
Loss function: MSE
Number of epochs:500
Batch size: 8

Learning rate: 0.0001
Shuffle training data in dataloader

Loss Curve:

The MSE of test set: 0.4028308093547821

TuneV1.6(Saved)

Normalization: SS
Layers: [11, 10, 10, 1]
Without dropout
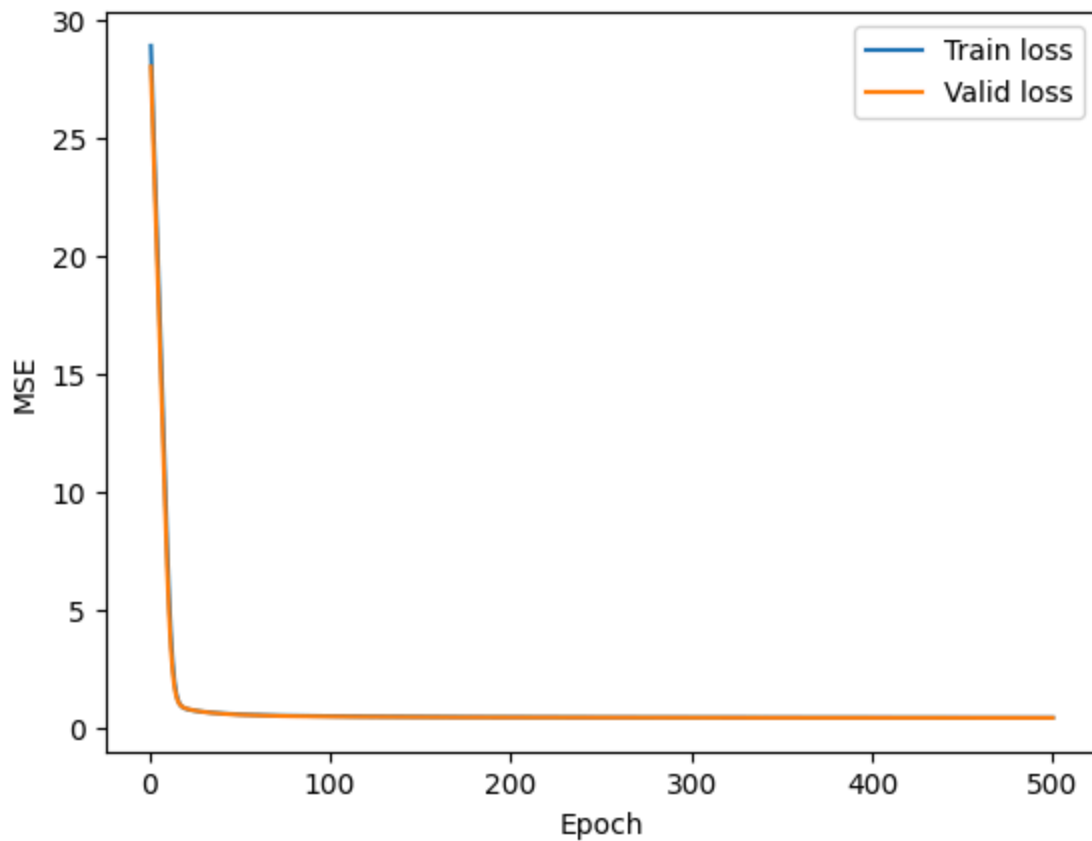Activation function: reLu
Loss function: MSE
Number of epochs:500
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss Curve:

The MSE of test set: 0.4162111282348633

TuneV1.7/V1.8

Normalization: SS
Layers: [11, 10, 10, 1]
Dropout_rate: 0.1/0.5
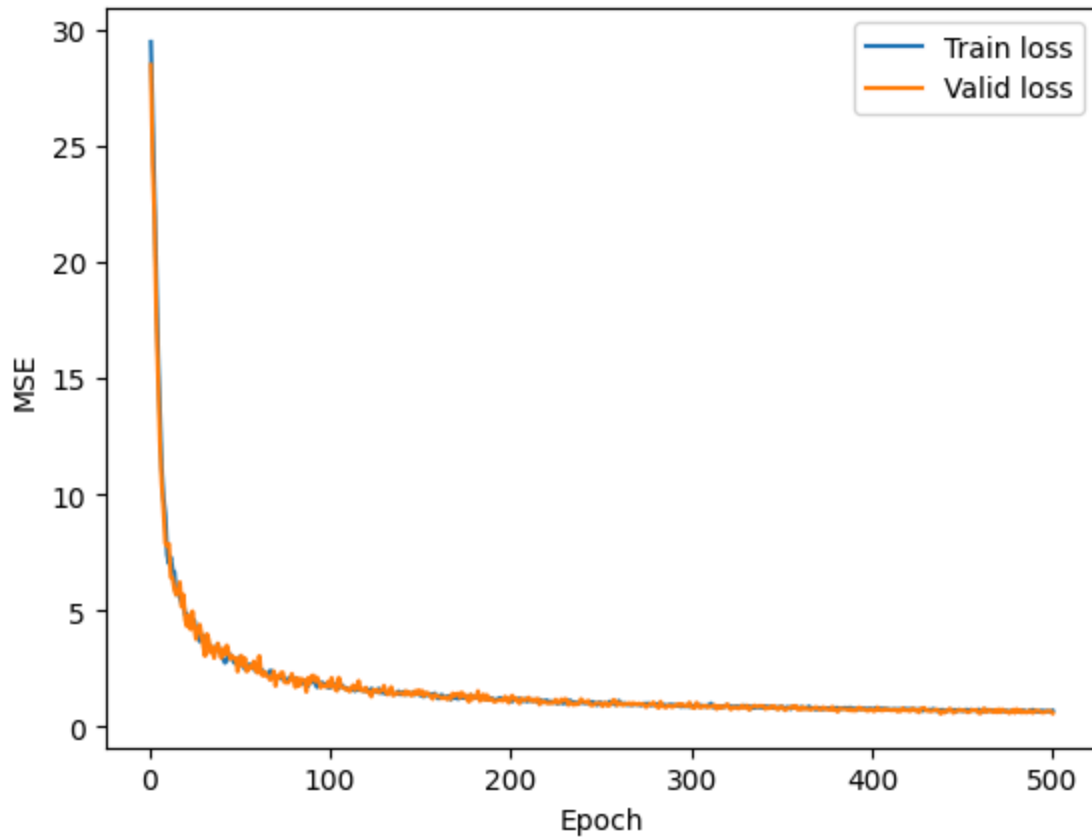Activation function: reLu
Loss function: MSE
Number of epochs:500
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss Curve (V1.8):

(V1.7)The MSE of test set: 0.6362356543540955
(V1.8)The MSE of test set: 0.7410240173339844

Dropout will make the performance worse.

TuneV1.9/V1.10(Saved)

Normalization: SS
Layers: [11, 10, 10, 1]
Dropout_rate: 0
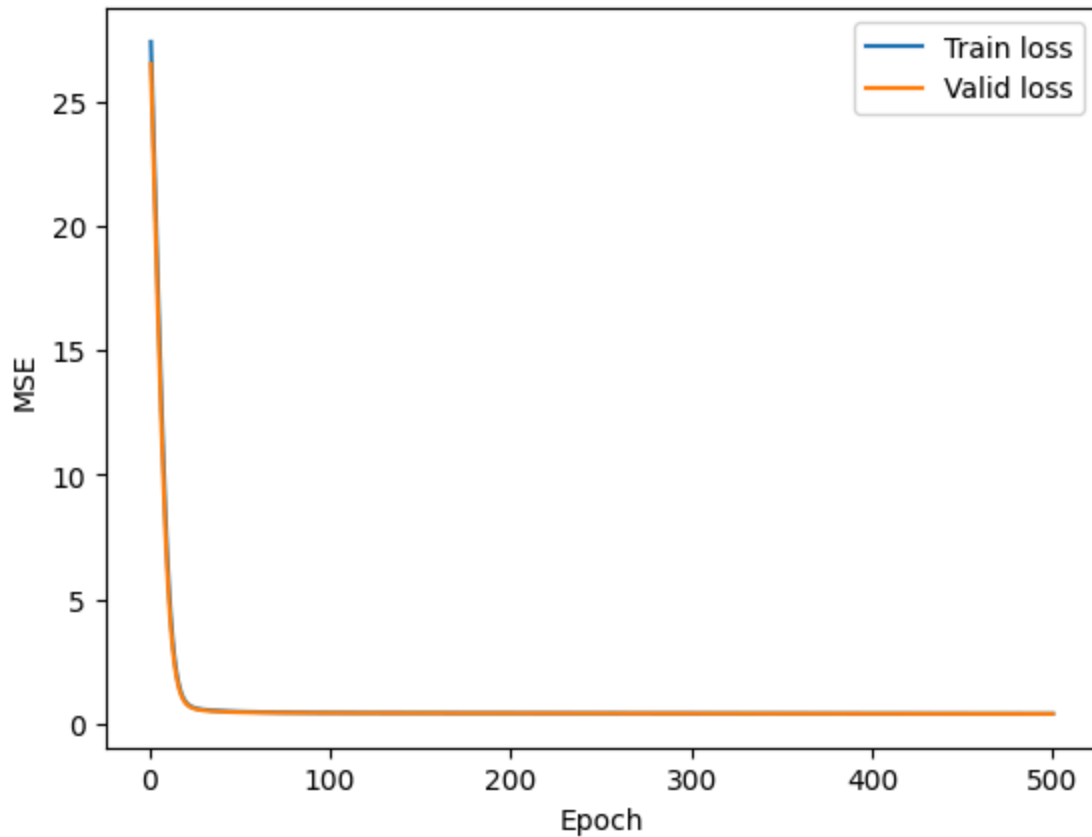**Activation function: Sigmoid/Tanh**
Loss function: MSE
Number of epochs:500
Batch size: 8
Learning rate: 0.0001
**Shuffle training data in dataloader**

Loss Curve(V1.10 Tanh):

(Sigmoid)The MSE of test set: 0.5983479619026184
(Tanh)The MSE of test set: 0.41961660981178284

TuneV2 (Selected from the best models from V1.*)

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu
Loss function: MSE
Number of epochs:==5000==
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss curve:
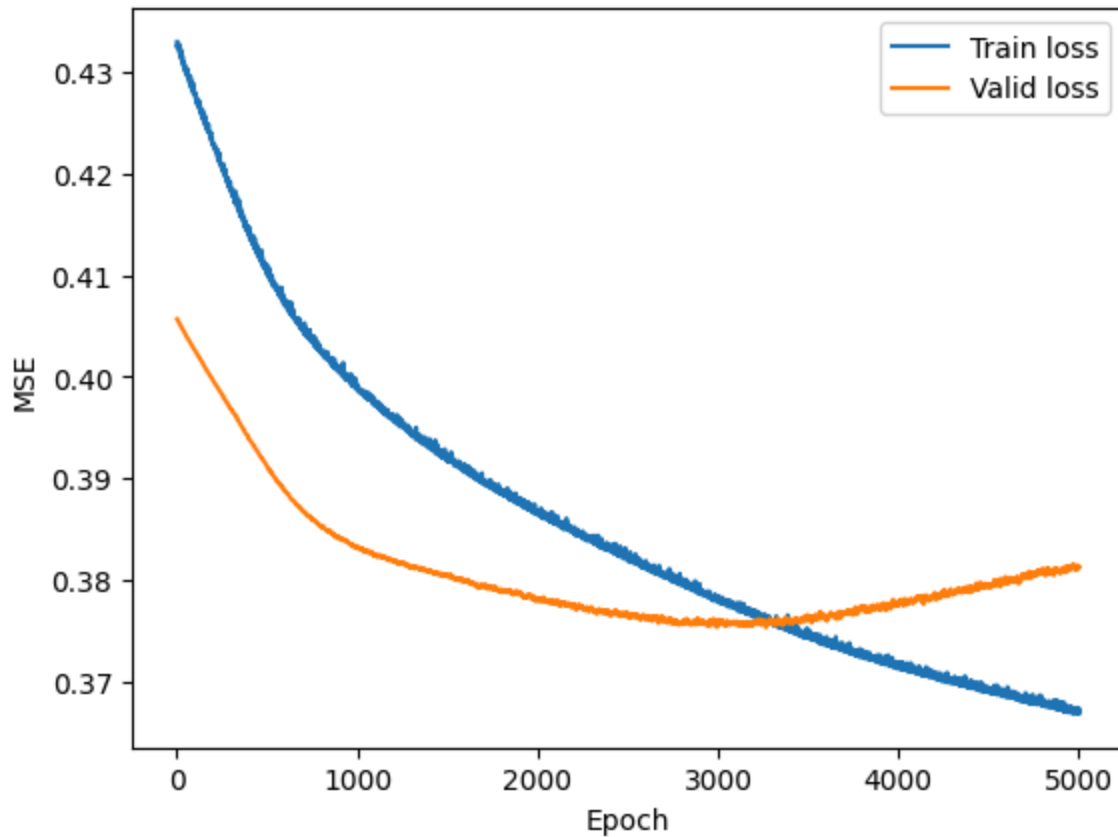
The MSE of test set: 0.40147849917411804

All right, time to set up early stopping

TuneV2.1(Saved)

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
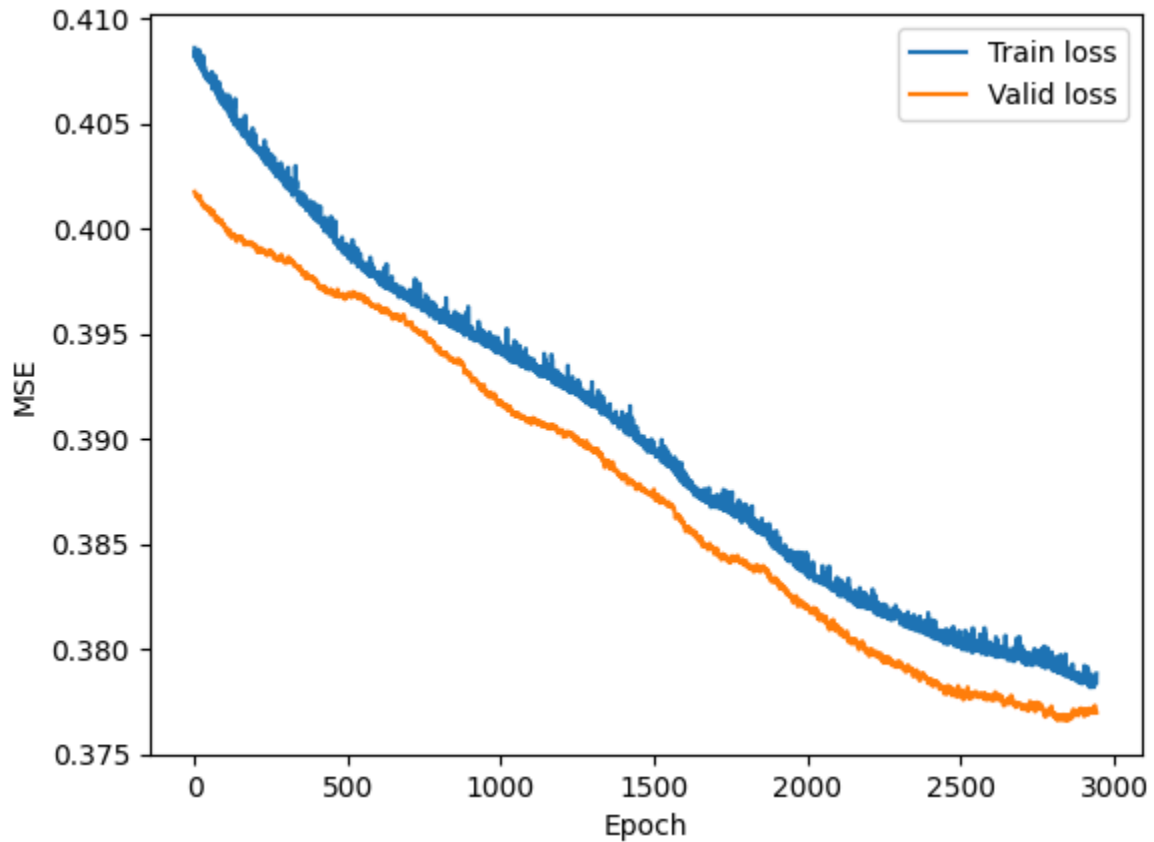**Activation function: Relu**
Loss function: MSE
Number of epochs: <mark>**5000 - early stop 100 (early stopped at epoch 2942)**</mark>
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader

Loss Curve:

The MSE of test set: 0.3814166486263275

TuneV2.2/TuneV2.3

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu
Loss function: MSE
Number of epochs: 5000 - early stop 100
Batch size: 32/16
Learning rate: 0.0001
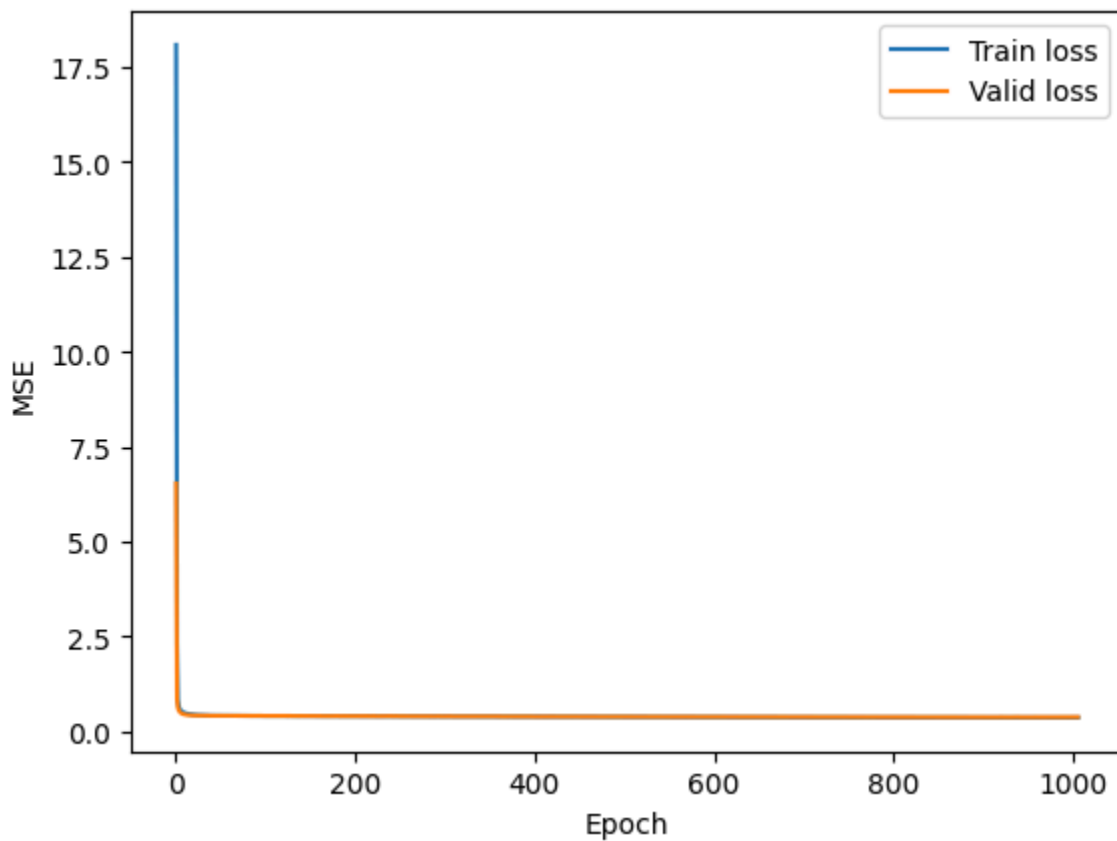**Shuffle training data in dataloader**

(V2.2)The MSE of test set: 0.40
(V2.3)The MSE of test set: 0.40603408217430115

Increasing Batch size is not improving the proformance

TuneV2.4(Saved)

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu
Loss function: MSE
Number of epochs: 5000 - early stop 100
Batch size: 8
Learning rate: 0.001
Shuffle training data in dataloader

Loss Curve:

The MSE of test set: 0.39196497201919556

TuneV2.5/TuneV2.6

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu

Loss function: MSE
Number of epochs: 5000 - early stop 100/20
Batch size: 8
Learning rate: 0.001
Shuffle training data in dataloader
momentum=0.5

(V2.5)The MSE of test set: 0.396
(V2.6)The MSE of test set: 0.38899001479148865

TuneV2.7(Saved)

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu
Loss function: MSE
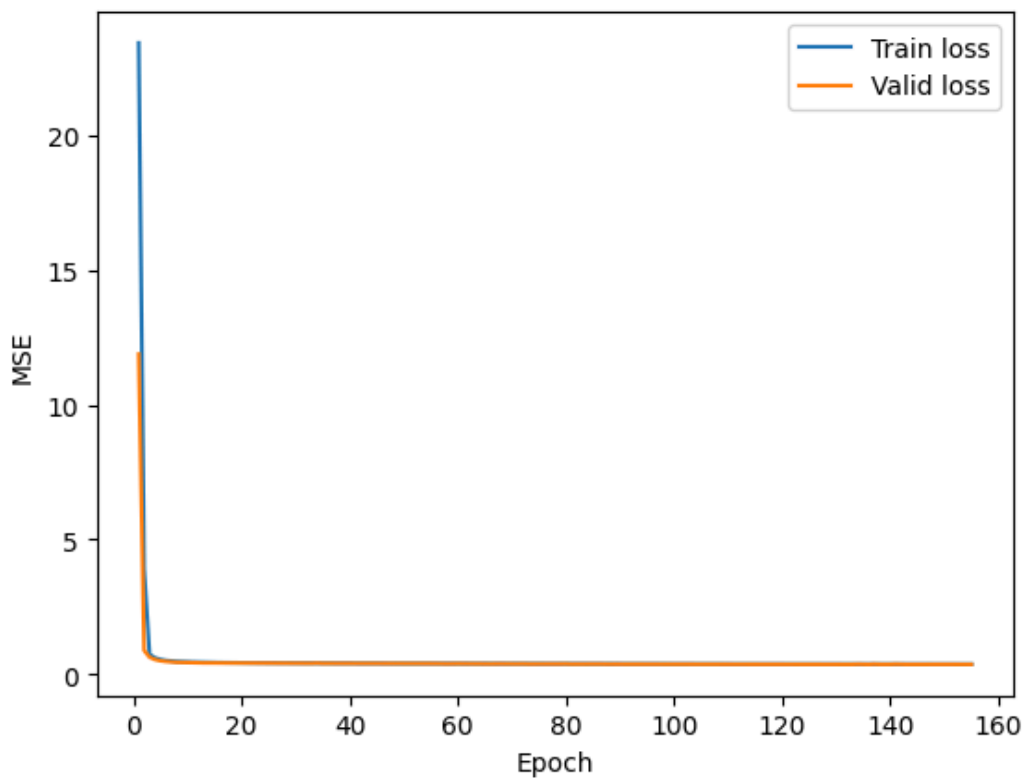**Number of epochs: 5000 - early stop 20**
Batch size: 8
Learning rate: 0.001
Shuffle training data in dataloader
momentum=0.2

Loss Curve

The MSE of test set: 0.37462902069091797

TuneV2.8

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu
Loss function: MSE
Number of epochs: 5000 - early stop 20
Batch size: 8
Learning rate: 0.001
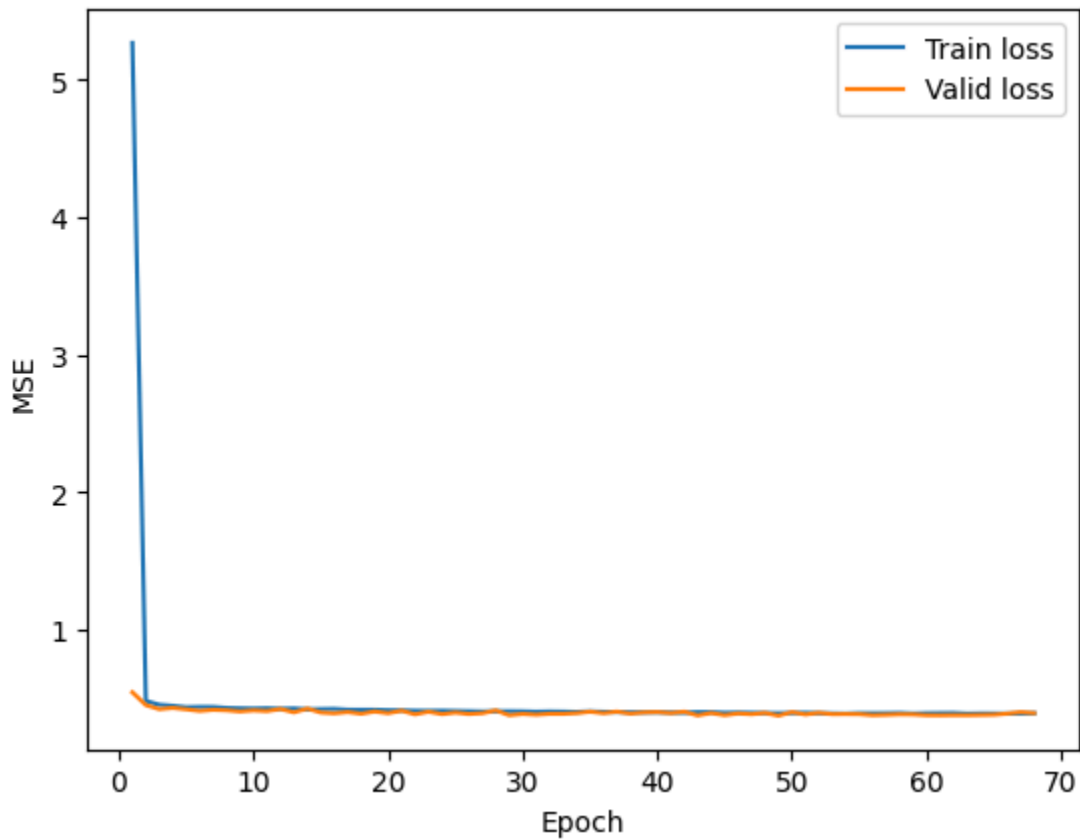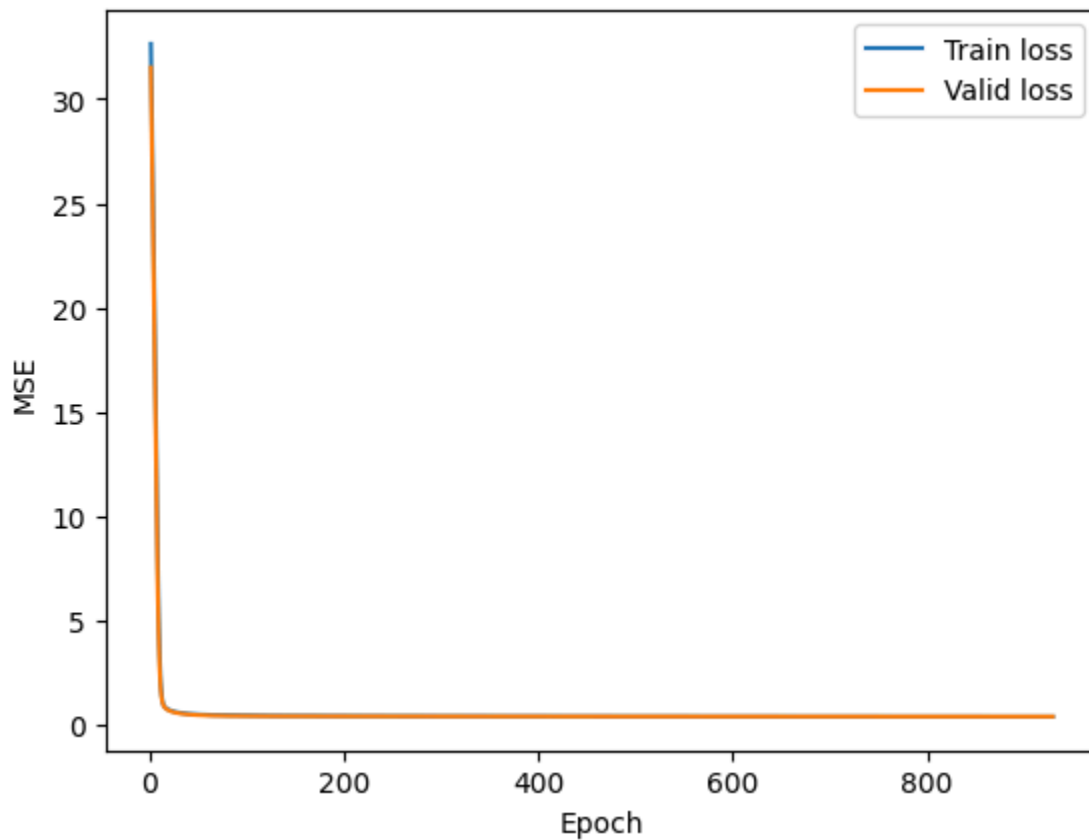**Shuffle training data in dataloader**
<mark>momentum=0.8</mark>

Loss curve:

The MSE of test set: 0.4131854176521301

TuneV2.9 (Saved)

Normalization: SS
Layers: [11, 5, 5, 1]
Dropout_rate: 0
Activation function: Relu
Loss function: MSE
Number of epochs: 5000 - early stop 50 (Stopping at epoch 929)
Batch size: 8
Learning rate: 0.0001
Shuffle training data in dataloader
momentum=0.2

Loss curve:

The MSE of test set: 0.38019827008247375

All right, I think the tuning is good enough!